

Configuración de la capa YARN

Archivo mapred-site.xml

En Hadoop se puede especificar el framework que queremos utilizar en MapReduce. Utilizaremos el valor "Yarn" en la propiedad "mapreduce.framework.name" para indicar a MapReduce que se ejecutará como una aplicación Yarn. Si no ponemos nada, el valor por defecto de esta propiedad es "local" indicando que todo el proceso de MapReduce se debe hacer en el nodo.

También modificamos la propiedad "mapreduce.application.classpath" para indicar la ruta donde están las librerías con las que se implementa MapReduce en un sistema distribuido. Como mínimo es necesario indicar las rutas de los archivos jar contenidos dentro de la carpeta "share/hadoop/mapreduce" dentro de la estructura de carpeta de hadoop. Debemos tener en cuenta que las rutas pueden ser diferentes por estar usando un usuario distinto o tener hadoop en alguna otra ruta.

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>

  <property>
    <name>mapreduce.application.classpath</name>
    <value>/home/administrador/hadoop/share/hadoop/mapreduce/*,/home/administrador/hadoop/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```

Para realizar esta configuración modificamos el archivo mapred-site.xml dentro de la carpeta de configuración de Hadoop.

```
nano a/home/administrador/hadoop/etc/hadoop/mapred-site.xml
```

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>

  <property>
    <name>mapreduce.application.classpath</name>
    <value>/home/administrador/hadoop/share/hadoop/mapreduce/*,/home/administrador/hadoop/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```

Archivo yarn-site.xml

El archivo yarn-site.xml se encuentra dentro de la carpeta de configuración de Hadoop. Al ejecutar MapReduce de manera distribuida, entre la parte Map y la parte Reduce, es necesario reordenar las salidas de los mappers y repartirla entre los reducers. De esta parte se encargará "mapreduce_shuffle" y se lo indicaremos a través de la propiedad "yarn.nodemanager.aux-services".

También tendremos que especificar las rutas de las clases Java que se utilizarán para poder ejecutar los mappers, los reducers y dar soporte a todo el framework MapReduce. Lo haremos bajo la propiedad "yarn.application.classpath" poniendo como valores las carpetas que contienen los archivos jar separadas por coma.

```
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>

<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>nodo01</value>
</property>
```

Para realizar esta configuración modificamos el archivo mapred-site.xml dentro de la carpeta de configuración de Hadoop.

```
nano /home/administrador/hadoop/etc/hadoop/yarn-site.xml
```

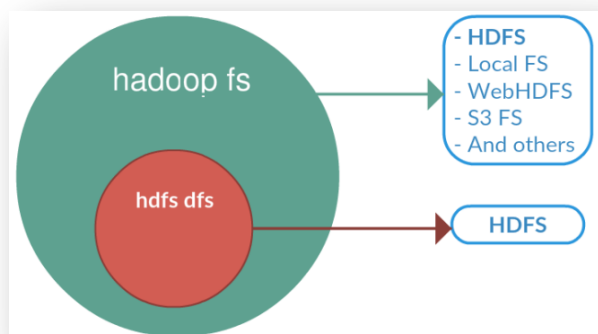
```
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>equipo01</value>
</property>

<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>

</configuration>
```

Puesta en marcha de la capa YARN

Antes de iniciar la capa de procesamiento YARN es necesario tener configurado y funcionando la capa de almacenamiento. Durante este curso trabajaremos exclusivamente con HDFS pero también es posible usar S3 de Amazon como alternativa privada, Ceph como otra posibilidad opensource a HDFS o, el propio sistema de ficheros local como alternativa para procesamiento local



Podemos iniciar la capa YARN de varias maneras. La primera opción es arrancar cada servicio de manera individual en cada nodo mediante el comando.

```
administrador@equipo01:~$ jps
5221 Jps
administrador@equipo01:~$ hdfs --daemon start namenode
administrador@equipo01:~$ jps
5340 Jps
5261 NameNode
administrador@equipo01:~$
administrador@equipo01:~$ yarn --daemon start resourcemanager
administrador@equipo01:~$ jps
5382 ResourceManager
5610 Jps
5261 NameNode
administrador@equipo01:~$
```

```
Hhfs --daemon start namenode
yarn --daemon start nodemanager
```

En este caso hay que tener claro qué funciones queremos que desempeñen cada uno de los módulos de clúster. En esta captura podemos ver como al hacer el primer "jps" no se muestra ningún servicio, arrancamos el namenode de HDFS y volvemos a comprobar con otro "jps" que se está ejecutando. En ese momento que ya sabemos que la capa de almacenamiento funciona, arrancamos el "Resource Manager" de YARN y comprobamos que ha arrancado correctamente. En los nodos esclavos tendremos que arrancar el servicio "Datanode" para HDFS y "NodeManager" para YARN.

Otra opción es arrancar en todos los nodos los servicios YARN que correspondan usando un script desde el nodo principal. El nodo adecuado para lanzar este script es el nodo principal porque aquí estará el archivo workers donde se consultará la lista de nodos esclavos. Debemos recordar que es necesario tener funcionando la parte de almacenamiento.

```
start-dfs.sh
```

```
administrador@equipo01:~$ start-dfs.sh
Starting namenodes on [equipo01]
Starting datanodes
Starting secondary namenodes [equipo01]
administrador@equipo01:~$
administrador@equipo01:~$
administrador@equipo01:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
administrador@equipo01:~$ █
```

```
start-yarn.sh
```

La tercera alternativa es usar otro de los scripts disponibles que, en este caso, se encarga de levantar la parte HDFS y la parte YARN en todos los nodos. Al igual que en el caso anterior, el mejor nodo para lanzar este script es el nodo principal ya que ahí tendremos el archivo workers con la lista de nodos

```
administrador@equipo01:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as administrador in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [equipo01]
Starting datanodes
Starting secondary namenodes [equipo01]
Starting resourcemanager
Starting nodemanagers
administrador@equipo01:~$
```

esclavos.

```
start-all.sh
```

En este caso ya no nos tenemos que preocupar de la capa de almacenamiento, se encarga el script.

Para parar los servicios podemos proceder de la misma manera sin más que sustituir el "start" por un "stop" en todos los comandos anteriores.

Otros aspectos para tener en cuenta.

En la configuración de YARN hay muchas propiedades que podemos modificar. Buscando el nombre del archivo de configuración en la web de hadoop encontraremos una tabla con todas las propiedades, valores por defecto y una pequeña ayuda de qué es lo que hace y valores admitidos. Especialmente cuando virtualicemos y/o tengamos un entorno pseudo-distribuido puede ser adecuado modificar parámetros relacionados con el tamaño y capacidad máxima de los contenedores.

Generalmente al virtualizar no creamos máquinas especialmente potentes ni en almacenamiento ni, especialmente, en memoria ni procesadores. Esto puede generar algún problema al ejecutar aplicaciones pesadas que soliciten un contenedor grande. Nuestros nodos virtualizados no tendrán capacidad y pueden fallar.

Por lo tanto, de la misma manera que hemos bajado el número de réplicas de 3 a 2 para economizar espacio en nuestra virtualización, también deberíamos ajustar el límite máximo de los contenedores.

La manera de trabajar con YARN es a través de MapReduce. La responsabilidad del programador se centra en la etapa de map donde se transforman los datos de entrada y en la etapa de reduce donde se acaban de procesar para obtener un resultado final. Entre estas dos etapas tiene lugar una etapa intermedia llamada shuffle en la que los datos de salida de los mappers se ordenan y se agrupan antes de pasárselos a los reducers. En la configuración que te propongo en esta guía hemos indicado que de esta etapa se encargue "mapreduce_shuffle" de YARN. Si queremos programar esa función será necesario cambiar esa propiedad.

Otra propiedad importante es en la que se hace referencia a las rutas de las librerías que se usan en MapReduce. Las que están en esta configuración hacen referencia exclusivamente a lo relacionado con MapReduce. Habría que plantearse como añadir las librerías que necesitemos en nuestro código de los mappers y los reducers. Debemos pensar que el código se va a ejecutar en cualquiera de los nodos por los que todos deben tener acceso a las librerías.

En este punto ya podemos ver que la configuración de Hadoop en general se basa en editar archivos XML para personalizar los valores de todas las propiedades. En muchas de estas propiedades hay que indicar rutas y esto es muy susceptible de cometer errores en la escritura. Estos errores son difíciles de detectar. La manera que propone hadoop para minimizar en lo posible estos errores es el uso de variables de entorno que apunten a rutas específicas. Durante este curso hemos configurado las variables mínimas necesarias que son JAVA_HOME y HADOOP_HOME. En principio hadoop será capaz de llegar hasta sus archivos con solo conocer el valor de HADOOP_HOME y desde allí, anexar rutas relativas. Si nos documentamos en la web podremos ver como en muchas otras configuraciones hacen un uso

mucho más habitual de las variables. En este curso he preferido siempre escribir las rutas absolutas desde la raíz del sistema de ficheros para que quede absolutamente claro de donde están los archivos. En caso de querer aprovechar la facilidad y limpieza que proporciona el uso de variables de entorno te propongo que investigues el archivo “hadoop-env.sh” que encontrarás dentro de la carpeta de configuración de hadoop. Si consideras interesante usar variables también se pueden declarar a mano. En este caso el archivo preferido para hacerlo es en “/etc/environment” ya que son comunes para todos los usuarios configurados en ese nodo y además no depende de iniciar sesión con ningún shell en concreto.

De esta manera no será necesario loguearse en los nodos y será suficiente con simplemente tenerlos encendidos para que el clúster pueda funcionar. Pensemos que si declaráramos las variables en el archivo “.bashrc” de un usuario, estas solo estarían en memoria una vez que el usuario se ha logueado y mantiene la sesión activa.

En definitiva, en esta guía se ha optado por un enfoque más didáctico que “limpio”. La ventaja es que estamos preparados para interpretar cualquier escenario en producción que nos encontremos.